Abstract—This paper describes a novel methodology for high-level control and coordination of autonomous vehicle formations and its demonstration on high fidelity models of the Organic Air Vehicle developed at Honeywell Laboratories. The scheme employs decentralized receding horizon controllers that reside on each vehicle to achieve coordination among team members. An appropriate graph structure describes the underlying communication topology between the vehicles. On each vehicle, information about neighbors is used to predict their behavior so that conflict-free trajectories that maintain coordination and achieve team objectives result. When feasibility of the decentralized control is lost, collision avoidance is ensured by invoking emergency maneuvers that are computed via invariant set theory.

I. INTRODUCTION

Interest in the coordination and control of Unmanned Air Vehicles (UAVs) has grown significantly over the last years. The main motivation for this trend is the wide range of military and civilian applications where teams of these vehicles working together have the potential of providing a low cost and efficient alternative to existing technology.

From a control engineering perspective, formation flight can be seen as a large control problem in which the objective is to compute the inputs that drive the vehicles along trajectories which maintain relative positions as well as safe distances between each UAV pair. Several approaches to formation control have been proposed in the literature (see [1], [2] and references therein). Optimal control problem formulation has been one of the most used frameworks for tackling this problem (see [3] and references therein). In this framework, formation flight is formulated as a minimization of the error between UAV relative distances and desired displacements. Such a formulation allows collision avoidance requirements to be easily included as additional constraints between each UAV pair. Although centralized optimal or sub-optimal approaches have been used in different studies (see, for instance [3], [4]), it is clear that, as the number of vehicles increases, the solution of such large scale, centralized, non-convex optimization problems becomes prohibitive. This is true even when the most advanced optimization solvers and much simplified linear vehicle dynamics are used [5]. The main challenge is to formulate simpler decentralized problems which result in a formation behavior similar to what is obtainable with a centralized approach.

The scheme proposed in this paper presents a decentralized design based on a hierarchical decomposition. In such a decomposition, the lower level comprises the UAV dynamics equipped with efficient guidance and control loops. At the higher level, the controlled UAV can be represented sufficiently well as a constrained multi-input, multi-output (MIMO) linear system. For this class of systems, a decentralized optimization-based control framework is constructed [6]–[8]. Specifically, Receding Horizon Control (RHC) is utilized to achieve formation flight and other cooperative tasks. Within this framework, individual vehicles use neighbor information to predict their behavior in order to avoid collisions and act cooperatively. The proposed approach can handle constrained MIMO linear or piecewise linear models of unmanned vehicles. The optimization problem is formulated and solved as small Mixed-Integer Linear Programs (MILPs) which can be translated into equivalent gain scheduled controllers requiring significantly lower computational effort that enables real-time implementation. Furthermore, different maneuvering objectives such as formation keeping, formation splitting and joining are dynamically attained by changing appropriate terms in the cost function. The procedure is also robust to changing number of vehicles in the team since individual vehicles may rely on information from any number of neighbors in order to determine their control policies. These benefits are demonstrated on a high fidelity model of the Organic Air Vehicle (OAV) developed at Honeywell Laboratories in Minneapolis.

The paper is organized in two parts. First, the dynamics of the 29-inch OAV is briefly described in Section II, followed by its lower-level guidance and control design in Section III. Section IV gives an overview of the decentralized optimization-based approach used for higher-level coordinated controller synthesis. The second part of the paper demonstrates the application of these techniques to OAV formation flight, using simulation results carried out on high fidelity models of the vehicles in Section V.

II. OAV DYNAMICS

The Organic Air Vehicle (OAV) is a scalable autonomous ducted-fan vehicle which is being developed as a demonstrator vehicle to meet agility, ease of deployment and other requirements for future UAV systems. Because of its vertical take-off and landing capability as well as its relatively small
size, such a vehicle offers several potential tactical advantages for company- and other lower-level reconnaissance and surveillance activities. There are also numerous envisaged applications of smaller-scale versions of this vehicle for homeland security. Future applications could conceivably deploy hundreds of OAVs flying in formation to accomplish specific objectives.

The OAV exhibits a highly nonlinear, constrained multivariable character. In the following description, a brief overview of OAV modeling and low-level control is presented.

![Diagram of OAV](image)

Fig. 1. OAV modeling.

The OAV is depicted in Figure 1(a). The vehicle translates and rotates by modulating thrust and propeller wash along its axis via the deflection of specific vane sets. For instance, to translate in a forward direction, the vehicle ‘tilts’ by an angle proportional to the desired speed of travel in the given direction. Tilt angles are similar to pitch and roll angle in the helicopter coordinate system notation. The body $z$ axis points forward in the duct inlet plane. This is also the direction in which a mounted camera would point. It lies along the hinge of vane sets that are used for rolling motion. The body $y$ axis points to the right in the duct inlet plane. This axis lies along the hinge axis of vane sets used to pitch the vehicle. The body $x$ axis points down along propeller axis.

The vehicle is modeled by (conceptually) breaking it up into its constituent parts and then developing physics-based models for each component. These building blocks are shown in Figure 1(b).

The four sets of vanes include left and right pitch vanes; and forward and aft roll vanes. In the dynamic model of the OAV, the inputs are the vane deflections and the thrust along the propeller axis. Aerodynamic and propulsion models that map these quantities to forces and moments on the vehicle’s rigid body are developed from first principles using the geometry of the surfaces and by treating the problem as a basic aircraft model.

III. LOW-LEVEL CONTROL DESIGN / HIGH-LEVEL MODELING

Nonlinear control of the ‘inner loop’ (i.e., attitude and rate loops) and the ‘outer loop’ (i.e., position and velocity loops) is accomplished via nonlinear dynamic inversion [9] and robust multivariable control. The nonlinearities of the various loops are cancelled (to a certain degree) by inversion and desired dynamics are imposed on the resulting system so that the behavior resembles a set of integrators. Due to imperfect inversion, the response of these state variables tend to be more similar to a low-order transfer function rather than a pure integrator.

The overall control scheme is depicted in Figure 2. This kind of structure has been implemented at Honeywell as a reusable set of flight control tools called Multi Application Controls – Honeywell (MACH). MACH combines nonlinear design methods, classical flight control structures and design parameters, and robust multivariable design principles. Apart from the OAV, it has been used successfully on several programs including the F-18 High Alpha Research Vehicle and is currently being applied in DARPA’s Micro Air Vehicles (MAV) program.

![Diagram of OAV control system](image)

Fig. 2. The controlled OAV model.

High-level commands to the MACH-controlled system take the form of desired $N, E, h$-positions and heading angles (Figure 2). The vehicle may also be commanded via way-points expressed in terms of desired positions/heading or corresponding velocities/heading rate to these coordinates. The position/velocity control system takes these position commands as inputs and generates corresponding tilt (pitch, roll) and throttle commands. The tilt and heading commands are the inputs to the attitude/rate control system. Its outputs are actual vane deflections required to accomplish the commanded maneuver. These vane deflections, as well as the throttle and wind disturbances are the inputs to the OAV model.

Assuming near-perfect dynamic inversion, the dynamics from the commanded position and heading to the outputs (which may be selected as actual positions and heading) is that of a multivariable linear system. Although imperfect dynamic inversion may introduce slight coupling among other state variables, the heading dynamics can be considered decoupled from the $N, E, h$ position. Therefore they are not considered in the remainder of the paper. In OAV formation flight, vehicles at the higher level can be modeled as a third order constrained linear MIMO system in each axis. The inputs to the system dynamics are commanded positions along the $N, E, h$-axes, and the outputs are positions and velocities along the $N, E, h$-axes.

The high-level closed-loop OAV dynamics are described...
using the following linear discrete-time model
\[ x_{k+1} = f(x_k, u_k), \]
\[ y_k = h(x_k), \]  \hfill (1)
where the state update function \( f : \mathbb{R}^9 \times \mathbb{R}^3 \rightarrow \mathbb{R}^9 \) and output map \( h : \mathbb{R}^3 \rightarrow \mathbb{R}^6 \) are linear functions of their inputs. The states and inputs of the vehicle at time \( k \) are denoted by \( x_k \in \mathbb{R}^9 \) and \( u_k \in \mathbb{R}^3 \), respectively.

The dynamical model (1) is constrained to directly account for practical limitations on the vehicles’ velocity and acceleration. It also allows redesign and modification of the inner-loop controllers to track velocity or acceleration references directly.

The simplified OAV dynamics in (1) were obtained using linear identification techniques based on position step responses of the closed-loop controlled nonlinear OAV model. The position response of the OAV can be considered decoupled and identical in both North and East directions. It is important to emphasize that the approach proposed in the following section can easily accommodate higher order, more complex linear or piecewise-linear models that describe the OAV dynamics with higher fidelity.

OAV autonomous formation flight is rendered tractable using high-level closed-loop vehicle dynamics (1) for formation control. A group of OAVs can be controlled by commanding either desired position or speed or even acceleration over a shifted prediction horizon.

### IV. High-Level Control Strategy

In this section, we introduce a decentralized Receding Horizon Control (RHC) scheme used for high-level control of the OAV. The main concept of RHC is to use a model of the plant to predict the future evolution of the system [10]. Based on this prediction, at each time step \( t \) a certain performance index is optimized under operating constraints with respect to a sequence of future input moves. The first of such optimal moves is the control action applied to the plant at time \( t \). At time \( t+1 \), a new optimization is solved over a shifted prediction horizon.

The decentralized RHC scheme is formulated for dynamically decoupled systems, such as OAVs flying in formation, which are coupled only by the cost function and constraints of an optimization problem. The main idea of the proposed framework is to break a centralized RHC problem into problems of smaller sizes [6]. Each RHC controller is associated with a different vehicle and computes the local control inputs based only on its states and that of its neighbors. On each vehicle, the current state and the model of its neighbors are used to predict their possible trajectories and move accordingly (similarly to what we do while driving cars). The information-exchange topology and inter-vehicle constraints are described by a graph structure in the problem formulation. The cost function will depend on the formation’s mission and include terms that minimize relative distances and/or velocities between neighboring vehicles. The coupling constraints arise from collision avoidance. The interaction graph is full (each vehicle has to avoid all the other vehicles) but it is approximated with a time-varying graph based on a “closest spatial neighbors” model. A more formal discussion of the proposed scheme follows.

#### A. Problem formulation and decentralized control scheme

Consider a set of \( N_v \) linear decoupled dynamical systems, where the \( i \)-th system is described by the discrete-time time-invariant state equation:
\[ x_{k+1}^i = f^i(x_k^i, u_k^i), \]
\[ y_k^i = h^i(x_k^i), \]  \hfill (2)
where \( x_k^i \in \mathbb{R}^{n_i}, u_k^i \in \mathbb{R}^{m_i}, f^i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \rightarrow \mathbb{R}^{n_i}, h^i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}^{p_i} \) are states, inputs, state update function and output function of the \( i \)-th vehicle, respectively. Let \( U^i \subseteq \mathbb{R}^{m_i} \) and \( Y^i \subseteq \mathbb{R}^{p_i} \) denote the set of feasible inputs and outputs of the \( i \)-th vehicle
\[ y_k^i \in Y^i, \quad u_k^i \in U^i, \quad k \geq 0, \]  \hfill (3)
where \( Y^i \) and \( U^i \) are given polytopes.

The set of \( N_v \) constrained systems will be referred to as the multi-vehicle system. Let \( \bar{x}_k \in \mathbb{R}^{N_v \times N_v} \) and \( \bar{u}_k \in \mathbb{R}^{N_v \times N_v} \) be the vectors which collect the states and inputs of the multi-vehicle system at time \( k \), i.e.,
\[ \bar{x}_k = \begin{bmatrix} x_{k,1} & \ldots & x_{k,N_v} \end{bmatrix}, \]
\[ \bar{u}_k = \begin{bmatrix} u_{k,1} & \ldots & u_{k,N_v} \end{bmatrix}, \]
\[ \bar{x}_{k+1} = f(\bar{x}_k, \bar{u}_k). \]  \hfill (4)

The equilibrium pair of the \( i \)-th vehicle is denoted by \( (\bar{x}_{e,i}, \bar{u}_{e,i}) \), and \( (\bar{x}_{e}, \bar{u}_{e}) \) denotes the corresponding equilibrium for the multi-vehicle system.

So far the individual vehicles belonging to the multi-vehicle system are completely decoupled. We consider an optimal control problem for the multi-vehicle system where cost function and constraints couple the dynamic behavior of individual systems. A graph topology is used to represent the coupling in the following way. The \( i \)-th system is associated to the \( i \)-th node of the graph. If an edge \((i, j)\) connecting the \( i \)-th and \( j \)-th node is present, then the cost and the constraints of the optimal control problem will have a component which is a function of both \( x^i \) and \( x^j \). The graph will be undirected, i.e., \((i, j) \in \mathcal{A} = (j, i) \in \mathcal{A}\) and furthermore, the edges representing coupling change with time. Therefore, before defining the optimal control problem, we need to define a graph (which can be time-varying)
\[ \mathcal{G}(t) = \{\mathcal{V}, \mathcal{A}(t)\}, \]  \hfill (5)
where \( \mathcal{V} \) is the set of nodes \( \mathcal{V} = \{1, \ldots, N_v\} \) and \( \mathcal{A}(t) \subseteq \mathcal{V} \times \mathcal{V} \) the set of time-varying arcs \((i,j)\) with \( i \in \mathcal{V}, j \in \mathcal{V} \).

Using a particular graph structure at some time instant \( k \), the optimization problem is formulated as follows. Denote with \( \bar{x}_k^i \) the states of all neighbors of the \( i \)-th vehicle at time \( k \), i.e.,
\[ \bar{x}_k^i = \{x_{k,j}^i \in \mathbb{R}^{n_v} | (j,i) \in \mathcal{A}(k)\}, \]
\[ \bar{u}_k^i = \{x_{k,j}^i \in \mathbb{R}^{n_v} | (j,i) \in \mathcal{A}(k)\} \]
\[ \bar{y}_k^i = \{x_{k,j}^i \in \mathbb{R}^{n_v} | (j,i) \in \mathcal{A}(k)\} \]
\[ \bar{y}_k^i = \{x_{k,j}^i \in \mathbb{R}^{n_v} | (j,i) \in \mathcal{A}(k)\} \]

Analogously, \( \bar{u}_k^i \in \mathbb{R}^{n_v} \) and \( \bar{y}_k^i \in \mathbb{R}^{p_v} \) denote the inputs and outputs to all the neighbors of the \( i \)-th vehicle at time \( k \), respectively. Let
\[ g^{ij}(x^i, x^j) \leq 0 \]  \hfill (6)
define interconnection constraints \( \mathcal{C} \) between the \( i \)-th and the \( j \)-th vehicle, with \( g^{ij} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_z} \). For formation flight, these constraints define non-convex collision avoidance requirements in the following way:

\[
g^{ij}(x_k^i, u_k^i, x_k^j, u_k^j) = d_{safe} - \| y_{k, pos}^i - y_{k, pos}^j \|_p. \tag{7}
\]

The parameter \( d_{safe} \) in constraint (7) represents a lower bound on the norm of relative position between neighboring vehicles. In this paper we will consider \( p = 1, \infty \) which lead to square protection zones around vehicles. In general, collision avoidance guarantees between any OAV pair in a formation flight problem would necessitate the use of a full graph for describing inter-vehicle constraints (OAV protection zones should not intersect for every vehicle pair). This would prevent the use of any approach but a centralized one. For practical decentralization purposes it is usually sufficient for each vehicle to consider only a neighboring subset of all vehicles to accomplish formation flight. Due to possible changes of the required formation, this subset is likely to change leading to a time-varying interconnection graph. Furthermore, the allowed number of vehicles in these neighboring subsets might be limited, leading to a directed interconnection structure and graph in (5). The proposed scheme can be naturally extended to cover these situations as well.

Consider the following cost

\[
l(\bar{x}, \bar{u}) = \sum_{i=1}^{N_v} l_k^i(x^i, u^i, \bar{x}_k^i, \bar{u}_k^i), \tag{8}
\]

where \( l_k^i : \mathbb{R}^{n_x} \times \mathbb{R}^{m_u} \times \mathbb{R}^{n_{\bar{x}}} \times \mathbb{R}^{n_{\bar{u}}} \rightarrow \mathbb{R} \) is the cost associated with the \( i \)-th vehicle at time \( k \) and is a function of its states and the states of its neighbor vehicles. Assume that \( l \) is a positive convex function with \( l(\bar{x}, \bar{u}) = 0 \). The cost function \( l_k^i \) in (8) is assumed to have the following form:

\[
l_k^i(x^i, u^i, \bar{x}_k^i, \bar{u}_k^i) = \underbrace{Q \begin{bmatrix} y^i - y_f^i \\ \bar{y}_k^i - \bar{y}_f^i \\ \Delta y_k^i - \Delta y_f^i \\ \Delta \bar{y}_k^i - \Delta \bar{y}_f^i \end{bmatrix}}_{p} + \underbrace{R \begin{bmatrix} u^i \end{bmatrix}}_{p}, \tag{9}
\]

where \( \Delta y_k^i \) is a stacked vector of relative outputs \( (y^j - y^i) \) for all \( j \) such that \( (i,j) \in \mathcal{A}(k) \). It represents the difference between the \( i \)-th vehicle outputs and its neighbors’ outputs. The variable \( \Delta y \) denotes a similar collection of relative outputs between the neighbors of the \( i \)-th vehicle, i.e., it is comprised of \( (y^j - y^i) \) output differences for all \( q, r \) such that \( (i,q) \in \mathcal{A}(k) \) and \( (i,r) \in \mathcal{A}(k) \). Subscript \( f \) denotes steady-state final reference values for all variables. The above general cost function involves weights on the states and control inputs of the \( i \)-th OAV and its neighbors, as well as weighing the relative states between them.

We formulate a decentralized RHC scheme based on the cost function (8), system models (2) and constraints (6). Given a certain graph interconnection structure \( \mathcal{G}(t) \), let the following finite time optimal control problem \( \mathcal{P}_i(t) \) be associated to the \( i \)-th system at time \( t \)

\[
\min_{\bar{u}_1^i(t)} \sum_{k=0}^{N-1} l_k^i(x_k^i, u_k^i, \bar{x}_k^i, \bar{u}_k^i) + l_N^i(x_N^i, \bar{x}_N^i), \tag{10a}
\]

subject to

\[
x_{k+1}^i = f^i(x_k^i, u_k^i, \bar{x}_k^i, \bar{u}_k^i), \tag{10b}
\]

\[
y_k^i = h^i(x_k^i), \quad k \geq 0 \tag{10c}
\]

\[
y_k^i \in \mathcal{Y}_k^m, \quad u_k^i \in \mathcal{U}_k^m, \quad k = 1, \ldots, N - 1 \tag{10d}
\]

\[
(i, j) \in \mathcal{A}(t), \quad k = 1, \ldots, N - 1 \tag{10e}
\]

\[
g_k^i(x_k^i, u_k^i, \bar{x}_k^i, \bar{u}_k^i) \leq 0, \quad (i, q) \in \mathcal{A}(t), \quad (i, r) \in \mathcal{A}(t), \tag{10f}
\]

\[
x_1^i = x_0^i, \quad \bar{x}_0^i = \bar{x}_0^i, \tag{10g}
\]

where \( \bar{U}_1^i \triangleq \{u_0^i, u_1^i, \ldots, u_{N-1}^i, u_N^i\} \in \mathbb{R}^s \), \( s = (m_i + m)N \) denotes the optimization vector, \( x_0^i \) denotes the state vector of the \( i \)-th vehicle predicted at time \( t+k \) obtained by starting from the state \( x_t^i \) and applying to system (2) the input sequence \( u_0^i, \ldots, u_{N-1}^i \). The tilded vectors denote the prediction vectors associated to the neighboring systems assuming a constant interconnection graph. Denote by \( \bar{U}_1^i = \{u_0^i, u_1^i, \ldots, u_{N-1}^i, u_N^i\} \) the optimizer of problem \( \mathcal{P}_i(t) \). The decentralized RHC scheme is defined as follows. At time \( t \)

1. Compute graph interconnections \( \mathcal{A}(t) \) according to a chosen policy.
2. Each vehicle \( i \) solves problem \( \mathcal{P}_i(t) \) based on measurements of its state \( x_t^i \) and the states of all its neighbors \( \bar{x}_k^i \).
3. Each vehicle \( i \) implements the first sample of \( \bar{U}_1^i \).

\[
u_1^i = u_0^i. \tag{11}\]

4. Each vehicle repeats steps 1 to 4 at time \( t+1 \), based on the new state information \( x_{t+1}^i, \bar{x}_{t+1}^i \).

In order to solve problem \( \mathcal{P}_i(t) \) each vehicle needs to know its current states, its neighbors’ current states, its terminal region, its neighbors’ terminal regions and models and constraints of its neighbors. Based on such information each vehicle computes its optimal inputs and its neighbors’ optimal inputs assuming a constant set of neighbors over the horizon. The input to the neighbors will only be used to predict their trajectories and then discarded, while the first component of the \( i \)-th optimal input of problem \( \mathcal{P}_i(t) \) will be implemented on the \( i \)-th vehicle. The solution of the \( i \)-th subproblem will yield a control policy for the \( i \)-th vehicle of the form \( u_1^i = c^i(x_t^i, \bar{x}_t^i) \).

A detailed discussion on feasibility and stability issues of decentralized RHC schemes goes beyond the scope of this paper. Some important observations can be found in [6], [11]–[13]. A detailed stability analysis of the decentralized RHC scheme presented in (10)-(11) is described in [7].

A schematic diagram of the hierarchical decomposition approach to OAV formation control is shown in Figure 3. Each OAV is controlled by its lower-level guidance and control loops, which receive commands from the on-board
decentralized RHC algorithm operating in closed-loop based on local measurements. In addition to local measurements, information (such as state estimates) is exchanged with neighbors. Exchange of other types of data is also allowed (e.g., predicted optimal solution sequence) if sufficient communication bandwidth is available. The mission manager is located on a ground station, which instructs the formation to perform certain mission-specific maneuvers by simply modifying their position and velocity reference values. The maneuver will be performed autonomously without interaction with the mission manager. The ground station is also capable of assigning an interconnection graph structure for the group of OAVs. Alternatively, the interconnection structure can be determined “on-the-fly” by the OAVs in a decentralized fashion based on, for instance, a closest spatial neighbor policy. There might be multiple on-board RHC controllers, which are generated using different cost functions and activated only in the corresponding phases of a mission. In the next section, we describe how collision avoidance can be ensured within the decentralized framework.

B. Guaranteed collision avoidance

As highlighted in Section IV-A the decentralized RHC problems are not guaranteed to be always feasible due to the mismatch between predicted and actual neighbor trajectories. This problem can be approached in different ways. In [6] collision avoidance guarantees are formulated in terms of robustness of the single decentralized scheme to prediction errors on neighbors’ trajectories. In [14], [15] inter-vehicle coordination (e.g. “right-of-way”) rules were established by means of including binary decision variables in the cost function or in the constraints of the local decentralized controllers. These rules have the ability to help in resolving conflicts between planned vehicle trajectories and reduce the likelihood of collisions.

For the purpose of OAV formation flight, we will rely on emergency controllers and their invariant sets to define protection zones and state constraints that guarantee collision avoidance when the local RHC subproblems become infeasible [14]. The basic idea is described next.

An emergency controller is defined for each OAV, which performs an emergency maneuver and controls the aircraft to a given reference if the feasibility of the local decentralized RHC problem is lost. For instance, the emergency controller can be designed to bring the vehicle to a full stop (e.g. in case of a hovering vehicle such as the OAV), or fly around in a circle (if using a fixed wing aircraft). The maximal positively invariant set of the closed-loop vehicle dynamics under emergency control respecting physical constraints can be calculated and used to aid in the selection of protection zone sizes around vehicles, which guarantee collision avoidance. In addition, more restrictive state constraints might also be necessary to guarantee that each vehicle is operated within this invariant set at any time during normal operation. Since the vehicle is always operated within the invariant set of the emergency controlled closed-loop system, switching to emergency control will always lead to collision-free trajectories.

If a linear state-feedback emergency controller is used with the discrete-time linear time-invariant system models and polyhedral state and input constraints, then the resulting closed-loop maximal positively invariant sets will be polyhedra. These sets can be easily computed off-line with simple techniques using polyhedral manipulations based on [16].

If the local RHC problem becomes feasible again, then normal operation resumes and the emergency control law is disactivated. A logic state variable $x_{t,L}^N$ is introduced for each OAV to indicate whether they are operating in normal or emergency mode. This information is transmitted to neighbors to aid in making more accurate predictions. Collision avoidance is achieved by augmenting the RHC scheme (10)-(11) with additional constraints based on the emergency invariant set. Further details of this modified decentralized RHC scheme with collision avoidance guarantees and possible issues are described in [14].

C. Real-time implementation

We refer to [7] for details on the use of equivalent PWA form of the decentralized RHC law.

V. HIGH-FIDELITY SIMULATION RESULTS

This section presents details of the decentralized control scheme (10)-(11) applied to the formation flight of OAVs flying at a certain altitude. The RHC control laws are implemented on the actual guidance and control loops and the high-fidelity nonlinear OAV dynamics based on wind-tunnel data.

The effectiveness of the proposed scheme is illustrated in a complex scenario for a group of six vehicles performing popup obstacle avoidance and formation joining and splitting. The interconnection graph will be time-varying in this case, and is obtained by having each OAV communicate with at most two of its closest neighbors. Depending on the line-of-sight obstruction caused by obstacles, the number of visible neighbors might be less than two at a certain time instant. This closest-neighbor interconnection policy means that the graph becomes directed.

The individual cost functions include terms that weigh the maximum control effort and the infinity norm of a vector which collects all the absolute and relative errors of the $i$-th OAV and its neighbors with respect to the final reference values. The interconnection constraints are defined as square
protection zones around vehicles that cannot intersect each other. The scaled separation distance was chosen to be $d_{safe} = 0.6$ in each spatial dimension. Solutions generated by each OAV enforce these collision avoidance constraints not only between itself and its neighbors, but among the neighbors as well.

The relative position and velocity reference values specified for each OAV correspond to a wedge vehicle formation. This formation was required to move with a common absolute velocity reference (2 ft/s in the $x$-axis). Absolute position references were not specified. Two popup obstacles appear during the simulation, at 3 and 9 seconds, respectively. The obstacles block communication links, which is reflected in the time-varying interconnection graph throughout the simulation. The formation is instructed by the mission manager to split into two subgraphs at 21 seconds into the simulation and move in different directions, by artificially restricting communication among vehicles in different groups and changing their absolute velocity references.

Figure 4 depicts eight snapshots of this scenario with six vehicles including popup obstacles, formation joining and splitting. Initially, six OAVs take off individually and hover in a straight line formation (Time Frame A). Then the process of reconfiguration into a wedge formation is started (Time Frame B). A popup obstacle is avoided by OAV #1 before the reconfiguration is completed (Time Frame C). All vehicles travel in the wedge formation for some time (Time Frame D). Another popup obstacle breaks the formation into two groups and therefore prevents communication among them (Time Frame E). Group A consists of OAVs #1, #5 and Group B includes OAVs #2, #3, #4, #6. After the popup obstacle is avoided, group members can communicate again and the wedge formation is re-established (Time Frame F). Another reconfiguration takes place as the OAVs split into two groups instructed by the ground station. OAVs #1, #2, #5 enter into a triangular formation while OAVs #3, #4, #6 establish a straight line formation (Time Frame G). At the end of the simulation, all vehicles land on the ground (Time Frame H). Full movies of this simulation example can be found at [17] along with other practical scenarios such as formation reconfiguration for stationary surveillance.

REFERENCES